Svedberg Open
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

An Extreme Outlier Handled Stochastic Random Under-Sampling Technique To Handle Class Imbalance In VAERS Dataset To Enhance Adverse Events Prediction

Vaishali Ravindranath[1], Dr. Sarojini Balakrishnan[2*]

[1]Research Scholar, Department of Computer Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, India email: 21phcsp010@avinuty.ac.in
[2]Assistant Professor (SG), Department of Computer Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, India, email: sarojini_cs@avinuty.ac.in

ABSTRACT
The rapid development and deployment of vaccination drives have been crucial in controlling the pandemic. However, monitoring and predicting adverse reactions following vaccination is essential to ensure public safety and maintain vaccine confidence. This study utilizes the Vaccine Adverse Event Reporting System (VAERS) dataset to predict the likelihood of mortality following COVID-19 vaccination, addressing the challenge posed by the highly imbalanced nature of the dataset. Given the disproportionate ratio of mortality outcomes in the data, traditional classification models show significant predictive bias towards the majority class. To mitigate this, under sampling technique was applied to balance the classes before model training, aiming to enhance the performance of the predictive models towards the minority class (Fatal event). After evaluation of random under sampler and Near-Miss under sampler and it was found that Random under sampler is simple and effective solution on larger datasets. Hence, as a further enhancement, a multi-objective stochastic search method combined with random under sampling was proposed to achieve better F1 score and class distribution. As the performance of prediction was affected by extreme outliers in the data, the proposed method was further enhanced with extreme outlier handling method with window of (0.05 to 0.25). The proposed method achieved an accuracy and F1 score in the range of (92-97%) with extreme outlier window (0.05). This work validates the hypothesis that presence of under-represented patterns would contributes to misclassification and affects the performance of the model confirming that our integrated outlier-handling approach effectively optimizes model robustness.

KEYWORDS: Stochastic search, under sampling, Pharmacovigilance, machine learning

INTRODUCTION

Vaccination based immunization has been seen as a successful way to ensure Global Health Security (GHS). According to the World Health Organization (WHO), vaccines are providing protection against more than 20 life-threatening diseases from pneumonia to cervical cancer. Since 2010, 116 countries have started immunizing their population with vaccines that were not previously administered. Day by day there are plenty of innovations happening in the vaccine space. In order to successfully administer them to people there is a need for collaboration in continuous development, deployment and monitoring. Monitoring these programs at large scale involves managing massive volumes of heterogenous real-world data, which has created necessity for automated, data-driven interventions. This is where Pharmacovigilance plays a pivotal role by providing insights on the post vaccination adverse events (AEs), understanding temporal efficiency of vaccines and analyzing the demand for early clinical intervention.

During pandemics like Corona Virus Disease 2019 global outbreak of the COVID-19 due to an immediate requirement of global immunization, rapid authorization and deployment of multiple vaccines was done in war mode. These vaccines have played a critical role in mitigating the spread of the virus and reducing mortality rates associated with the disease. However, like all medical interventions, vaccines can cause other

adverse reactions in some individuals, ranging from mild side effects to severe health outcomes, including death. Monitoring these adverse reactions is crucial not only for ensuring public health and safety but also for maintaining public trust in vaccination programs.

The Vaccine Adverse Event Reporting System (VAERS), established in 1990, plays a crucial role in pharmacovigilance by serving as an early warning system to detect potential safety issues in vaccines licensed in the U.S. Managed by the CDC and FDA, VAERS collects reports of adverse events following vaccination, helping to identify unusual patterns that warrant further investigation. Although VAERS relies on passive reporting, it is instrumental in monitoring vaccine safety and guiding public health responses, especially during large-scale immunization efforts (Kaur & Gupta, 2020).

The VAERS database collects data on adverse events following vaccination from various sources, including healthcare professionals, vaccine manufacturers, and individuals. As a passive reporting system, VAERS allows anyone to submit reports of adverse reactions, with healthcare providers being mandated to report specific events, and manufacturers required to report all adverse events they are aware of. The database includes detailed information such as vaccine type, patient demographics, timing of the adverse event, and the description of symptoms. Although VAERS does not establish direct causality between vaccines and reported events, it serves as a valuable resource for identifying patterns and potential safety issues. The data is publicly available, enabling researchers, public health officials, and the medical community to access, analyze, and use it for vaccine safety monitoring and research. This transparency facilitates the detection of rare or unexpected adverse events, driving further investigation and evaluation of vaccine safety (Al Khames Aga et al., 2021; Jeyanathan et al., 2020; Riad et al., 2021).

The data collection approach in VAERS presents several challenges for machine learning analysis:

Passive Reporting Bias: Since VAERS relies on voluntary reports, the data can be incomplete or biased causing data inconsistency. Some adverse events may be underreported, while others might be exaggerated, leading to an imbalanced dataset. This can skew machine learning models, making it difficult to draw accurate conclusions about vaccine safety.

Data Quality Issues: VAERS data often includes inconsistencies, missing values, and unstructured information. These quality issues make it challenging to create reliable machine learning models, as algorithms depend on clean and structured data for accurate predictions.

Non-Causal Data: VAERS does not determine causality between a vaccine and an adverse event, only reporting correlations. This lack of causality can mislead machine learning models, which may interpret correlated events as causal, leading to incorrect predictions or conclusions.

Class Imbalance: The data typically has a large number of records with no or mild adverse events and only a small number of severe cases. This class imbalance can cause machine learning models to be biased toward predicting the majority class (no adverse events), reducing the ability to accurately predict rare but significant events.

Unstructured Data: Much of the information in VAERS reports, such as symptom descriptions, is in unstructured text format. This requires advanced natural language processing (NLP) techniques, which can be complex and may not always capture nuances, further complicating machine learning analysis.

Algorithmic Limitations: Standard oversampling techniques introduce synthetic noise, while many undersampling approaches such as Near-Miss become computationally expensive due to the immense scale of the VAERS database.

To address these limitations, this research proposes a novel algorithm: Extreme Outlier handled Stochastic Random Under-Sampling (EOSRUS). Unlike traditional approaches, EOSRUS integrates a multi-objective stochastic search mechanism that iteratively optimizes the balance between majority class reduction and outlier preservation. By binning and removing extreme model outliers defined as misclassified samples falling outside specific probability thresholds our method enhances the precision of adverse event prediction while retaining the computational efficiency required for large scale deployment.

The contributions of this work are three fold:

The EOSRUS Algorithm: A robust, scalable undersampling framework that explicitly mitigates the extreme outlier issues found in standard random sampling.

An End-to-End Predictive Pipeline: An automated ML architecture that integrates structured and unstructured features (via TF-IDF and NLP) to classify vaccine adverse events with high precision.

Empirical Validation: A comprehensive evaluation against state-of-the-art baselines with PyCaret models, demonstrating that EOSRUS achieves superior F1-score performance, particularly when deployed with ensemble classifiers.

OBJECTIVE

The primary objective of this research is to develop a robust sampling strategy to mitigate class imbalance in the VAERS dataset, thereby enhancing the predictive efficacy of machine learning models. While class imbalance can be addressed through various techniques such as oversampling, undersampling, and balanced ensemble learning this study focuses on advancing undersampling methodologies. Undersampling is prioritized for its superior computational efficiency and scalability, which are critical when processing the high-volume, real-world data characteristic of pharmacovigilance. Specifically, this work introduces the 'Extreme Outlier Handled Stochastic Random Under-Sampling' (EOSRUS) algorithm, designed to improve F1-score performance by iteratively optimizing the balance between majority-class reduction and the preservation of representative, non-outlier data points.

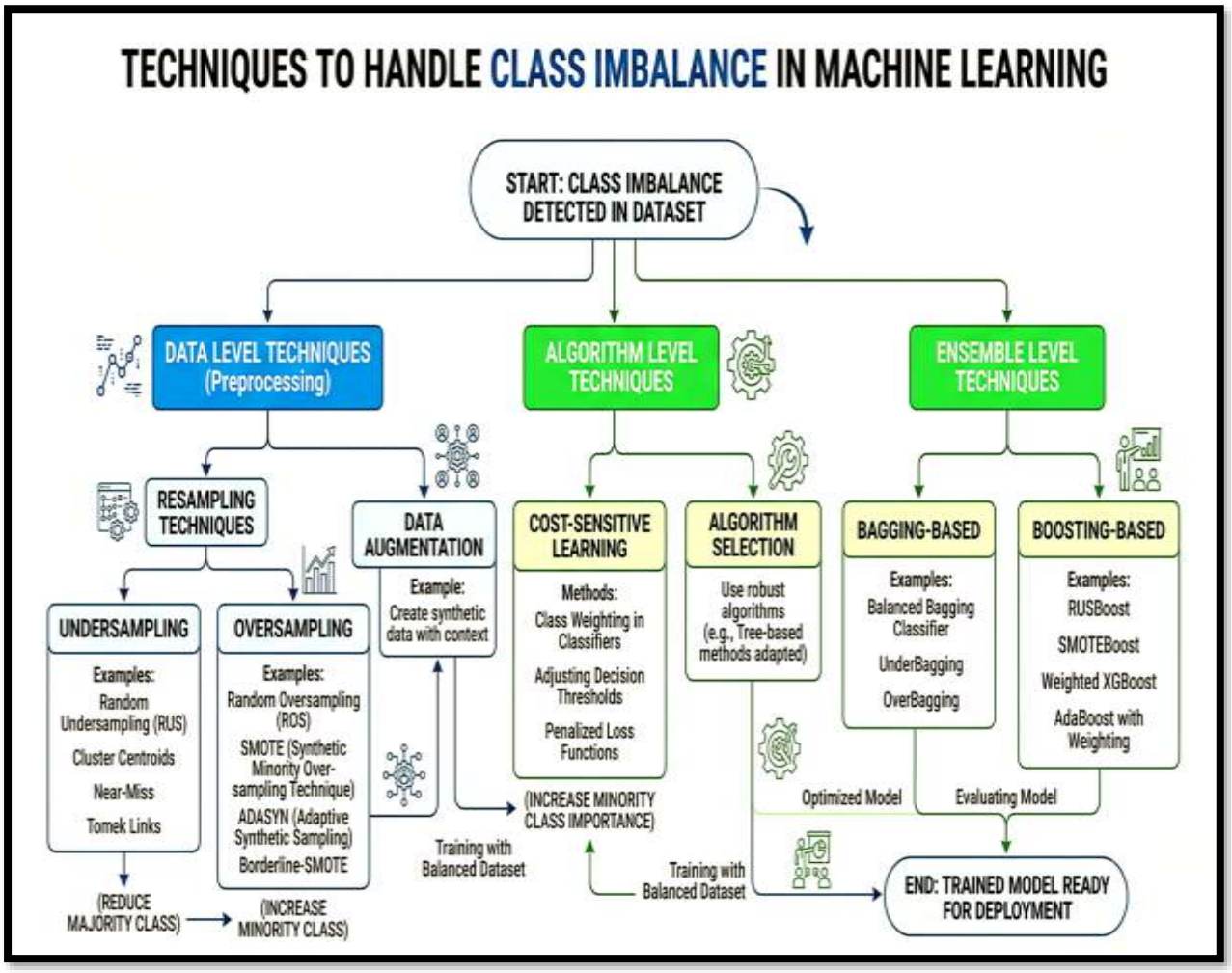


Fig 1. Classification of data resampling and algorithm level techniques for addressing data class imbalance in skewed datasets during machine learning

LITERATURE REVIEW

Numerous studies have investigated predictive analysis using the VAERS dataset. Depending on the methodological approach, most research has utilized machine learning and deep learning classifiers. Recent works started leveraging the advantages of transformer models and Language models in the predictive modeling space.

Early work employed classical machine learning classifiers on VAERS for adverse-event extraction. For instance, Saad et al. (2022) developed a voting-ensemble classifier (ER-VC) combining Extra-Trees and Logistic Regression to predict death risk post-COVID-19 vaccination, thereby improving soft-voting performance. Similarly, Botsis et al. (2011) utilized text-mining pipelines incorporating rule-based and machine learning classifiers (e.g., SVM, Random Forest) to automate anaphylaxis detection in H1N1 VAERS narratives. Subsequent research has expanded these methods to predict various adverse events, including morbidity, hospitalization, and mortality (Ahamad et al., 2022; Flora et al., 2022; Ravindranath & Balakrishnan, 2023). While machine learning (Gupta & Verma, 2023) and deep learning models (Du et al., 2021) have demonstrated utility in these predictions, challenges persist regarding class imbalance and the inherent noise in reporting data (Flora et al., 2022). Albayati and Altamimi (2023) reported that while Decision Tree models achieved an accuracy of approximately 91.3%, precision and recall were notably compromised by class imbalance.

To address the issue of class imbalance, Saad et al. (2022) employed oversampling methods such as SMOTE, which significantly improved accuracy to 98%. However, oversampling introduces synthetically generated scenarios that may deviate from the underlying data patterns; additionally, these methods increase dataset size by scaling the minority class to match the majority count, which can lead to computational inefficiency.

In another approach, Okamoto et al. (2023) applied Random Under-sampling with a Light Gradient Boosting Machine baseline to evaluate predictions against human-annotated responses. Their findings highlighted that an accuracy of 100% was unattainable due to the real-time bias inherent in machine learning models, reinforcing the necessity for explainable artificial intelligence.

The VAERS dataset contains mixed data types, including categorical, numerical, and free text. Techniques such as TF-IDF with NLP, Named Entity Recognition, and Bag-of-Words have been employed to transform this data into a tabular format (Wunnava et al., 2018). As summarized in Table 1, a diverse range of machine learning, deep learning, and generative AI models have been utilized for prediction. While generative AI models have demonstrated superior predictive performance compared to traditional machine learning, their "black box" nature remains a constraint regarding explainability. Although reasoning models are proposed as an alternative, they remain prone to hallucinations. Consequently, in clinical decision-making, the selection of explainable algorithms remains a high priority.

Table 1: Comparative Summary of VAERS-Focused Prediction Studies to systematically analyze literature on predictive modeling and preprocessing for the Vaccine Adverse Event Reporting System (VAERS), relevant studies are synthesized below. Table 1 provides a comparative overview of key methodological paradigms, study designs, performance metrics, and identified limitations or research gaps.

Approach Type	Study (Year)	Methodology	Key Results	Gaps / Limitations
Named-Entity Recognition	Du et al. (2021)	Ensembled CRF + Bi-LSTM + BioBERT + VAERS-BERT for NER of nervous system disorder entities on 91 annotated GBS-related influenza VAERS reports (1990–2016)	Ensemble micro-F1: 0.68 (exact/strict), 0.81 (lenient)	Focused on entity-level extraction for a single disorder (GBS/influenza); no report-level classification or class imbalance handling.
LLM-based Classifiers	Li et al. (2024)	GPT-2, GPT-3 variants, GPT-3.5, GPT-4, and LLaMA-2 evaluated (pretrained and fine-tuned) for NER-based AE extraction on same 91 VAERS reports; fine-tuned GPT-3.5 proposed as AE-GPT	AE-GPT (fine-tuned GPT-3.5) micro-F1: 0.704 (strict), 0.816 (relaxed)	Small dataset (91 reports); limited to GBS/influenza VAERS reports; LLMs are blackbox models.
Predictive Classification	Saad et al.	TF-IDF features + novel Extreme Regression Voting	Accuracy: 0.85, Precision: 0.85,	Synthetic samples require validation; no outlier

with Oversampling	(2022)	Classifier (ER-VC combining LR and Extra Trees) on COVID-19 VAERS data; SMOTE and ADASYN oversampling for multi-class death risk prediction	Recall: 0.85, F1: 0.84 (TF-IDF + SMOTE). ER-VC obtained accuracy 0.98.	handling and iterative model refinement.
Exploratory ML Analysis (no predictive classification)	Flora et al. (2022)	Unsupervised ML (association rule mining, self-organizing maps, hierarchical clustering, bipartite graphs) on 643,522 COVID-19 VAERS reports for AE pattern discovery; online survey of 211 participants	No classification metrics reported; identified top 20 AEs, cross-manufacturer patterns, and duplicate-removal impact (~7% variation)	Observational/exploratory only; no predictive model built; limited to structured VAERS fields; passive reporting biases unaddressed
Imbalanced Classification with Improved Undersampling	Chen et al. (2022)	Class-imbalance learning for predicting hospitalisation from AEFI (adverse events following immunisation) data; compared SMOTE, RUSBoost, and improved RUSBoost variants; features extracted from structured clinical and demographic fields	Improved RUSBoost achieved AUC 0.83, outperforming vanilla SMOTE and standard RUSBoost; deployed as a web-based clinical decision support tool	Random undersampling in RUSBoost discards majority samples indiscriminately without identifying hard-to-classify or overlapping boundary instances; no threshold-level outlier removal
Predictive Classification with Oversampling (Morbidity)	Ahamad et al. (2023)	Multiple ML classifiers (XGBoost, Random Forest, SVM, Logistic Regression, etc.) trained on VAERS COVID-19 data with medical history features; statistical feature selection; no explicit resampling — class imbalance addressed via algorithm-level cost sensitivity	Best classifier accuracy >90% for complication-free vaccination prediction; age, gender, comorbidities (diabetes, hypertension, heart disease) identified as top predictors	No explicit data-level resampling applied; severely imbalanced minority class (death/serious AE) not robustly handled; extreme outlier reports with ambiguous labels not removed prior to training
NLP + ML for Hospitalization Prediction	Tiwari et al. (2024)	NLP (Sentence-BERT embeddings) + ML classifiers (SVM, Random Forest, XGBoost) on VAERS COVID-19 narrative text to predict hospitalization; SMOTE applied for oversampling; TF-IDF and sentence embeddings compared	Best model (XGBoost + Sentence-BERT + SMOTE) achieved F1 ~0.87 for hospitalization prediction	SMOTE generates synthetic narratives from minority class without verifying clinical plausibility; misclassified borderline and extreme outlier reports remain in the training set, degrading decision boundary learning
Ensemble LLM + Deep Learning for AE Extraction	Li et al. (2025)	Ensembles of fine-tuned LLMs (GPT-2, GPT-3.5, GPT-4, LLaMA-2 7b/13b) and traditional deep learning models (RNN, BioBERT) for NER-based AE extraction from VAERS reports (n=621) and social media (Twitter, Reddit); evaluated on	Ensemble achieved strict micro-F1: 0.903; entity-level strict F1: 0.878 (vaccine), 0.930 (shot), 0.925 (AE)	Dataset heavily skewed toward non-AE tokens; rare and ambiguous AE mentions — particularly low-frequency severe outcomes — remain misclassified even in the best ensemble; no data-level resampling or outlier

		vaccine, shot, and AE entity types		filtering applied
Boundary-Aware Undersampling for Rare Event Detection	Abuzeid et al. (2026)	Progressive Clustering Undersampling (PCU) removes majority-class instances distant from minority-class boundary compared against 8 undersampling and 2 oversampling methods on highly imbalanced, noisy datasets; frequency-driven decision boundaries used	PCU consistently outperformed random undersampling, SMOTE, and other baselines on severely imbalanced and noisy data; two outputs optimised separately for F1 and precision	Validated on petroleum/anomaly detection datasets, not pharmacovigilance; extreme outlier instances (persistently misclassified across threshold iterations) not explicitly identified or removed before boundary sampling

Based on the gaps and findings from the literature, it is found that while scalable resampling techniques prioritize computational efficiency, they often lack the granularity to navigate the complex decision boundaries of highly imbalanced datasets. Conversely, sophisticated boundary-aware methods frequently impose huge computational costs when applied to the huge, real-world data like VAERS. Consequently, a critical research gap persists for a framework that is simultaneously scalable, robust to reporting noise, and sensitive to boundary-informative samples. This work addresses this research gap by introducing the Extreme Outlier Handled Stochastic Random Under-Sampling (EOSRUS) algorithm. By integrating a multi-objective stochastic search with probabilistic outlier binning, EOSRUS is designed to bridge this gap, providing a novel, computationally efficient solution that preserves the integrity of minority-class patterns while effectively mitigating the distortive influence of extreme outliers and reporting noise.

PROBLEM STATEMENT

Addressing class imbalance in the massive VAERS dataset requires a computationally efficient, scalable strategy. While oversampling relies on generating synthetic data which may misrepresent underlying patterns, undersampling preserves actual data by reducing the majority class to match the minority class. However, existing undersampling methods present a trade-off: boundary-aware techniques like Near-Miss are computationally expensive for large-scale datasets, whereas simple random undersampling is fast but vulnerable to noise and outliers. Consequently, there is a critical need for a robust undersampling framework that maintains the computational speed of random undersampling while incorporating an intelligent mechanism to filter out outliers and prevent biased sample selection. This research addresses that need by designing a method that balances efficiency with predictive accuracy.

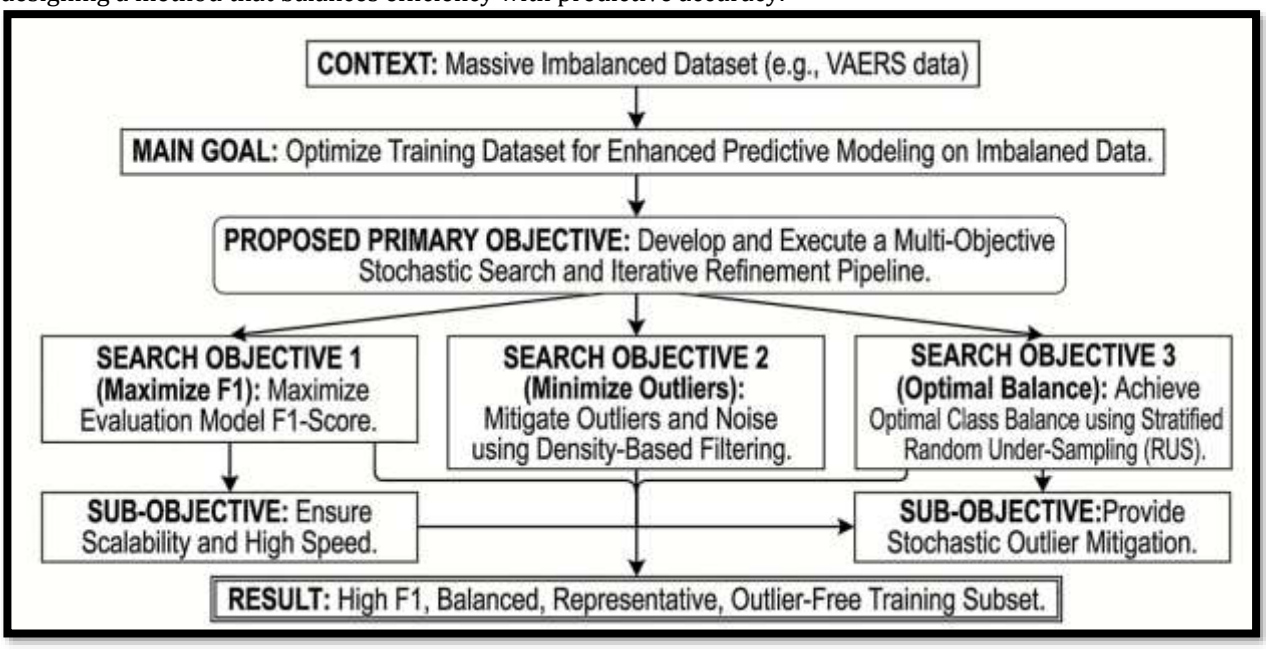


Fig 2. The objective behind the proposed improved Random under-sampling technique for class imbalance handling.

Proposed methodology

An improved random under-sampling strategy is proposed, in which a multi-objective stochastic search is utilized to enhance machine learning performance. The iterative refinement logic of Recursive Feature Elimination (RFE)—in which features are systemically reduced based on predictive thresholds—is adopted, and this principle is extended to the instance space through the iterative refinement of the training subset. To mitigate biased sample selection, a stochastic search is employed across T=100 iterations. Each random subset is evaluated by a multi-objective fitness function, through which higher F1-scores, outlier minimization, and optimal class balance are prioritized. An extreme outlier binning module is integrated into this process. The classifier's predicted probability scores are examined against ground-truth labels; for a binary classification, it is assumed that a sample labeled 'Y' should ideally score near 1.0, and 'N' near 0.0. When window size is set as 0.05, the misclassified instances with prediction scores within the confidence window [0.05, 0.95] are retained, while the extreme 0.05 tails which represent samples consistently misclassified by the model are excluded. By pruning these high-confidence outliers, noisy instances are removed, and the distortion of the decision boundary is prevented.

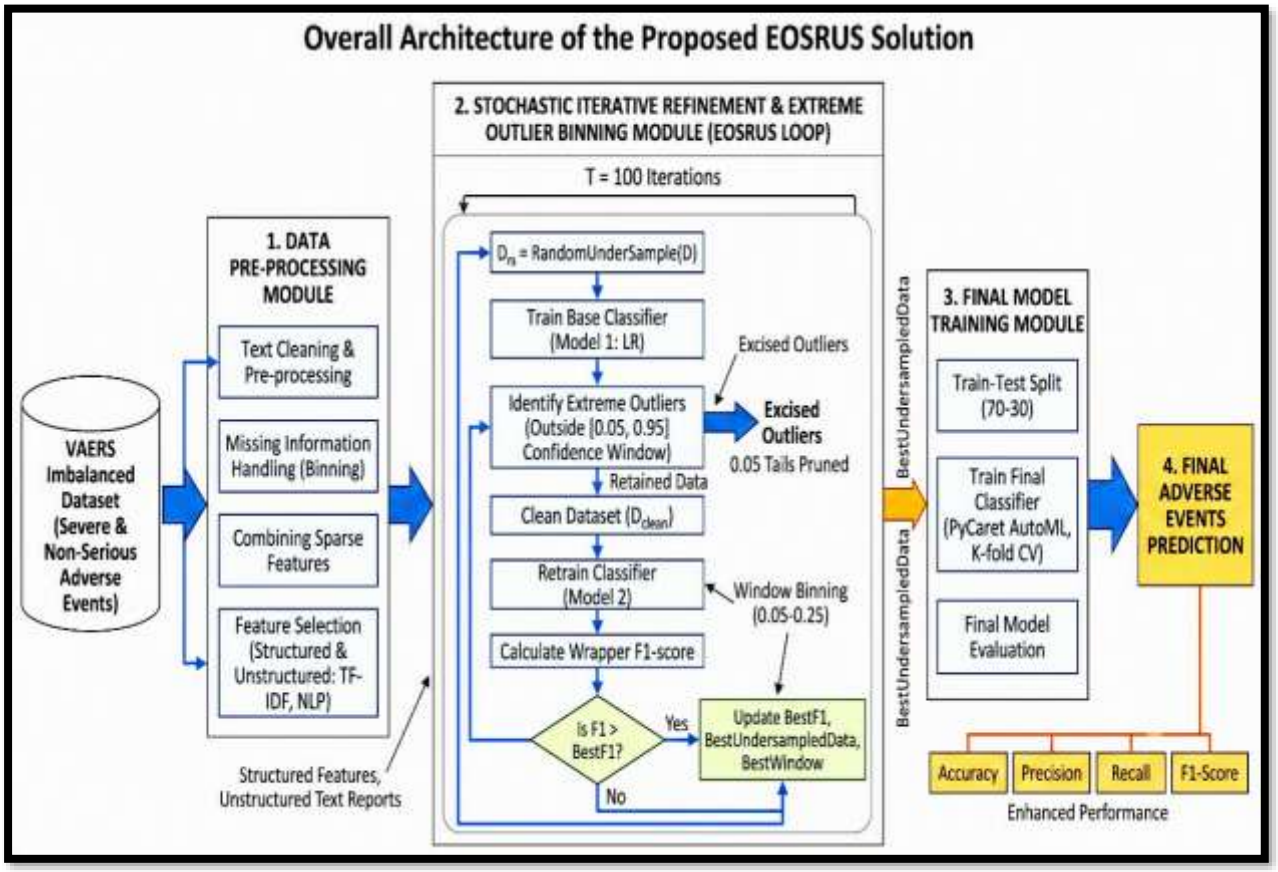


Fig 3. The overall architecture of the proposed under sampling solution EOSRUS under sampling algorithm

The overall architecture of the proposed EOSRUS solution is illustrated in Fig. 4. The system is structured into four distinct modules: Data Pre-processing, Stochastic Iterative Refinement, Extreme Outlier Binning, and Final Model Training. In the Data Pre-processing module, raw VAERS records are structured and cleaned to facilitate feature extraction. The architecture's novelty is centered within the Stochastic Iterative Refinement loop. In this stage, random subsets of the majority class are sampled across T iterations. A base classifier is trained on these subsets to facilitate Extreme Outlier Binning, where probabilistic prediction scores are compared against ground-truth labels. Instances falling outside the defined confidence window [W] are flagged as extreme outliers and excised. Finally, the subset that yields the highest F1-score is identified through the multi-objective fitness function and is established as the optimal training dataset for the final predictive model.

Proposed Algorithm | Extreme Outlier Handled Stochastic Random Under-sampling (EOSRUS)

```
Input: dataset D, iterations T=100,
    windows W=[0.05,0.10,0.15,0.20,0.25], classifier: LR

BestF1 = 0
BestWindow = None
BestUndersampledData = None

For each w in W:

  For t = 1 to T:

    D_rs = RandomUnderSample(D)        // balance classes

    Model1 = TrainClassifier(D_rs)

    Outliers = []

    For each (x_i, y_i) in D_rs:

      p = Model1.predict_proba(x_i)

      If y_i == 'A' and p[B] >= 0.5 + w:
        Outliers.append((x_i, y_i))

      Else if y_i == 'B' and p[B] <= 0.5 - w:
        Outliers.append((x_i, y_i))

    D_clean = D_rs without Outliers

    Model2 = TrainClassifier(D_clean)   // retrain on cleaned data

    F1_score = WrapperF1(Model2, D_clean)
                          // evaluate candidate subset

    If F1_score > BestF1:
      BestF1 = F1_score
      BestWindow = w
      BestUndersampledData = D_clean

Return BestWindow, BestF1, BestUndersampledData

// Final model evaluation
Split BestUndersampledData into D_train (70%) and D_test (30%)

FinalModel = TrainClassifier(D_train)

FinalPredictions = FinalModel.predict(D_test)

FinalF1 = ComputeF1(D_test.labels, FinalPredictions)

Return BestWindow, BestF1, BestUndersampledData, FinalF1
```

Experimental setup

The experimental setup is based on a Python working environment, utilizing the PyCaret framework for handling machine learning classification models. PyCaret's automated machine learning (AutoML) capabilities simplify the process of training multiple models and evaluating their performance.

Dataset Description:

The Vaccine Adverse Event Reporting System (VAERS) dataset found on Kaggle is a public, open-access resource that compiles self-reported post-vaccination adverse events in the United States. VAERS is

maintained by the CDC and FDA to monitor vaccine safety, and is updated regularly to capture potential signals from both routine immunizations and large-scale vaccination campaigns, such as COVID-19.
The dataset contains the following feature categories

- VAERSDATA: Contains core report metadata for each event (age, sex, location, vaccination date, onset date, vaccine manufacturer, outcomes such as death, disability, recovery, hospitalizations, etc.).
- VAERSSYMPTOMS: Lists up to five coded signs, symptoms, or diagnoses per report, using MedDRA terms.
- VAERSVAX: Provides details of the vaccines administered in each report, including manufacturer, dose, lot number, and type.

Typically, serious adverse events (death, life-threatening condition, disability, hospitalization, or congenital anomaly) constitute about 5–15% of all VAERS submissions, while the remaining 85–95% are non-serious or mild adverse events. This severe skew means that standard classification models tend to be biased toward predicting the majority (non-serious) class unless techniques such as under-sampling, oversampling, or cost-sensitive learning are applied.
Data Pre-processing: The dataset is first pre-processed, ensuring clean, structured data with balanced or under-sampled classes to address the class imbalance issue. The EOSRUS algorithm is used to iteratively sample subsets of the data for training, ensuring an optimal balance of diversity, low outlier presence, and class ratio.
Machine Learning Models: PyCaret package is used to train the following classification models, all using default hyperparameters provided by PyCaret by disabling auto pre-processing. These models are run on both the original and under-sampled datasets for comparison.

Evaluation Metrics: After training, the models are evaluated using common classification metrics. These metrics help in assessing the model's ability to generalize across different classes, especially for imbalanced datasets. The metrics include:

Metric	Formula	Description
Accuracy	Accuracy=(TP+TN)/(TP+TN+FP+FN)	Measures the overall correctness of the model by computing the ratio of correctly predicted instances to the total number of predictions.
Precision	Precision=TP/(TP+FP)	Represents the number of true positive predictions out of all positive predictions, highlighting the model's ability to avoid false positives.
Recall	Recall=TP/(TP+FN)	Measures how well the model captures the actual positives in the dataset, useful in understanding the model's ability to minimize false negatives.
F1 Score	F1=(2×Precision×Recall)/(Precision+Recall)	The harmonic mean of precision and recall, providing a balanced measure when precision and recall are not equally important. Ideal for imbalanced datasets.

Evaluation Process: The best subset of data, identified from the iterative process, is fed into each classification model. Each model's performance is assessed using the above metrics. Comparisons are drawn between models using these metrics to determine the most suitable classifier for the undersampled dataset. The model with the highest F1 score is considered optimal due to the focus on both precision and recall.

Iterative Process: The iterative undersampling algorithm is tested by varying N (number of iterations) to assess its impact on the final sample's quality.The performance of models trained on these subsets is compared against models trained on the full dataset to assess the effect of undersampling on classification performance.

Cross-Validation: Each model is subjected to k-fold cross-validation (default k=10 in PyCaret) to ensure that performance metrics are not overfitted to the training set and generalize well to unseen data. This helps in obtaining a robust estimate of model performance. For each fold, the training and validation sets are split

randomly, and the same evaluation metrics (accuracy, precision, recall, and F1 score) are computed. The mean values of the evaluation metrics across all folds are recorded to summarize the model's overall effectiveness.

Model Tuning: After evaluating the default performance, PyCaret's auto-tuning capabilities can be used to fine-tune hyperparameters of the best-performing models, such as adjusting the number of trees for Random Forest, boosting parameters for XGBoost, or changing the kernel type for SVM. This tuning helps optimize model performance further and is repeated over the best subsets generated by the iterative undersampling algorithm.

RESULTS AND EVALUATION

In the study, two key experiments were conducted to assess the effectiveness of under -sampling techniques for handling imbalanced datasets. First one was to compare the performances of Random Undersampling vs. Near Miss Undersampling to evaluate their impact on model performance when addressing data imbalance.

Random Undersampling (RUS): The key advantage of RUS is its simplicity and computational efficiency. This method Randomly subsample the majority class so that the number of majority samples matches (or approaches) the number of minority samples.

Let $D = D_{maj} \cup D_{min}$, D_{maj} is the majority-class subset and D_{min} is the minority-class subset.

$|D_{maj}| = N_{maj}$, and $|D_{min}| = N_{min}$

The random under-sampled set $D_{maj}^{(rand)}$ is obtained by:

$$D_{maj}^{(rand)} = \textbf{RandomSample}(D_{maj}, N_{min}) \quad (1)$$

The final balanced dataset is:

$$D' = D_{maj}^{(rand)} \cup D_{min} \quad (2)$$

where $|D_{maj}^{(rand)}| = |D_{min}|$

Near Miss Undersampling: This method selects majority class samples that are closest (in terms of distance) to the minority class, preserving samples that are more challenging to classify.

Let each majority sample x_i in D^{maj} have K-Nearest Neighbors in the minority class. Compute the average distance:

$$\bar{d}(x_i) = \frac{1}{k}\sum_{j=1}^{k} \quad \|x_i - NN_{min}^{(j)}(x_i)\| \quad (3)$$

where $NN_{min}^{(j)}(x_i)$ is the j^{th} nearest minority neighbor to (x_i).

Select the N_{min} majority samples with the **lowest** $\bar{d}(x_i)$:

$$D_{maj}^{(NM)} = \{x_i \mid x_i \in D_{maj},\} \quad (4)$$

Final dataset:

$$D' = D_{maj}^{(NM)} \cup D_{min} \quad (5)$$

Both under-sampling methods yield a balanced dataset, but Near miss is boundary-aware, while random under-sampling is agnostic to sample location. This makes the algorithmic Near miss lesser scalable than

Random Under-sampling. Random Under-sampling works irrespective of the data types whereas Near Miss requires the inputs to be numeric to compute K-NN Calculations.

In Table 1, the performance of both the Near miss and Random Under-sampling in the vaccine adverse event (survival) was conducted and tabulated.

Table 2. Comparison of performance of Near miss (NM) and Random under-sampling (RUS) algorithm

Model	Accuracy_Nm	Accuracy_RUS	Recall_Nm	Recall_RUS	Prec_Nm	Prec_RUS	F1_nm	F1_RUS
Ada Boost Classifier	0.86	0.85	0.85	0.87	0.86	0.84	0.85	0.85
Decision Tree Classifier	0.86	0.85	0.85	0.86	0.87	0.84	0.86	0.85
Extra Trees Classifier	0.87	0.85	0.86	0.87	0.87	0.84	0.86	0.85
Extreme Gradient Boosting	0.87	0.86	0.86	0.89	0.87	0.84	0.87	0.86
Gradient Boosting Classifier	0.86	0.86	0.87	0.89	0.85	0.83	0.86	0.86
K Nearest Neighbors Classifier	0.84	0.84	0.81	0.86	0.86	0.82	0.84	0.84
Light Gradient Boosting Machine	0.86	0.86	0.86	0.89	0.87	0.84	0.86	0.86
Linear Discriminant Analysis	0.85	0.85	0.84	0.87	0.86	0.83	0.85	0.85
Logistic Regression	0.85	0.85	0.85	0.87	0.86	0.84	0.85	0.85
Naive Bayes	0.66	0.82	0.33	0.92	0.96	0.77	0.49	0.84
Quadratic Discriminant Analysis	0.67	0.73	0.42	0.98	0.87	0.65	0.55	0.78
Random Forest Classifier	0.87	0.85	0.86	0.87	0.87	0.84	0.87	0.85
Ridge Classifier	0.85	0.85	0.84	0.87	0.86	0.83	0.85	0.85
SVM - Linear Kernel	0.85	0.85	0.83	0.86	0.86	0.84	0.85	0.85

Considering the F1 score, it is evident that the under-sampling improves the prediction scores by handling class imbalance. While comparing the historical performance of vaccination adverse events, it is noticed that there is no significant improvement happening in terms of accuracy after under-sampling. Moreover the performance of Near miss algorithm in Table 1 and random under-sampling algorithm appears almost similar. In Fig.2, the Area under the curve scores reiterates it. Hence for large datasets like VAERS, the

Random under-sampling is found as advantageous due to the ease of scalability and random selection without bias.

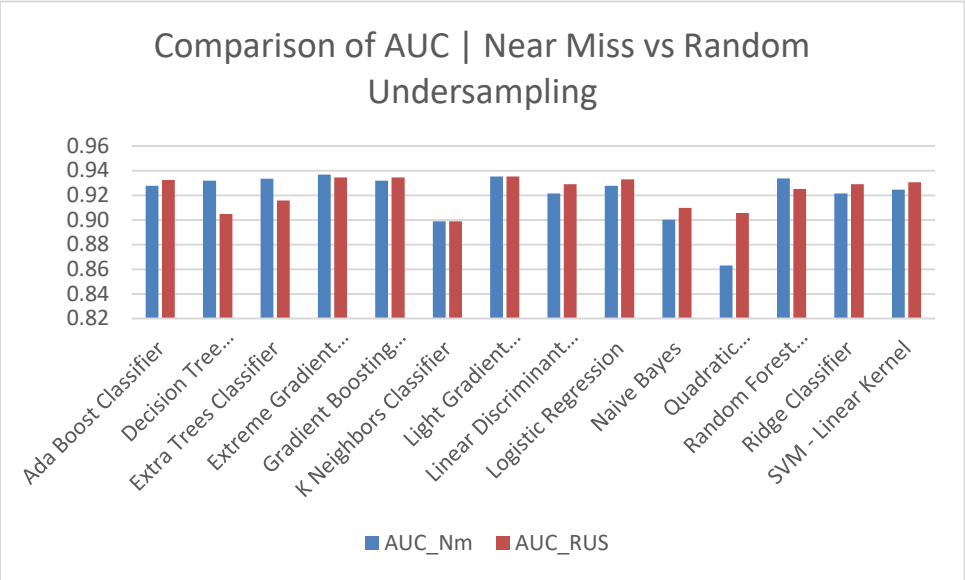


Fig 4. Comparison of Area Under the curve across different classifiers after Near miss and Random under-sampling

While choosing random samples, there could be under representation of certain class level data patterns in the dataset. These under represented samples are contributing to misclassification and creates confusion to the model in learning other related patterns. As discussed in the proposed model section, these noisy samples are considered as outliers. In this experiment, it was assumed that handling the extreme outliers with distribution rebalancing or binning would improve the performance of the model. An extreme outlier binning module is attached the proposed Stochastic random under-sampling algorithm to remove these noisy samples. In Experiment 2 the Random Under-sampling is compared with the proposed Extreme Outlier Handled Stochastic Search-based Under-sampling. The goal of the proposed method is to retain the speed of Random Under sampling while improving the quality of the selected samples to optimize performance. This approach iteratively selects random subsets of the majority class, but the selection is guided by the improvement in F1 Score described previously.

The fitness function considers F1 score to improve the overall quality of the training data. By using multiple iterations (controlled by the parameter N), the method explores various sample subsets and selects the one with the best fitness score, ensuring the sample is diverse and representative of the original dataset.

Evaluation: After running the stochastic search for multiple iterations, the best subset is identified and used to train the machine learning models. These models' performance is compared against the performance of models trained on the randomly under sampled dataset. Accuracy, Precision, Recall, and F1 Score are calculated to assess the impact of this improved approach on both the minority and majority class. The key comparison is whether the stochastic search-based method improves the F1 score and other metrics compared to simple Random Under sampling, indicating better handling of data imbalance with reduced bias and enhanced minority class representation.

The size of the VAERS Dataset taken is 659,946. This size is achieved after handling noise, ambiguity and redundancy in the raw data. The distribution of positive class of the target 'Survival Status' is skewed compared to the negative class. This distribution happens to be skewed always because in reality, the ultimate aim of mass vaccinations is to protect people from lethal adverse events. Hence, while under-sampling only (3-4%) of samples were retained by the random under-sampling algorithms. There are 33 columns in the data which comprises of symptoms, age group and historic factors converted into numeric values by TF-ID and one-hot encoding methods. The ambiguity in the data was handled using traditional NLP techniques. Through frequency based feature selection top 5 symptoms were chosen among the obtained symptoms to avoid noise due to data sparsity.

As discussed in the pseudocode, the EOSRUS works with a windowing function that bins the extreme model outliers. According to the binning window ranges [0.05,0.10,0.15,0.20,0.25], there was a difference in the number of training and test samples used in the model training and evaluation. Still, the distribution of class is not considerably affected as shown in Table 2. Often the test data mentioned in Table 2 is confused with the test data used during the EOSRUS undersampling. The transformed training and testing data are splitted from the undersampled data which is obtained after applying RUS and EOSRUS on the actual VAERS data. By this way the data leakage issue was handled in the experiment.

Table 3. The distribution of data and splits in various windows of EOSRUS is tabulated.

Description	**RUS**	**EOSRUS (0.05%)**	**EOSRUS (0.10%)**	**EOSRUS (0.15%)**	**EOSRUS (0.20%)**	**EOSRUS (0.25%)**
Target type	Binary	Binary	Binary	Binary	Binary	Binary
Selected columns	33	33	33	33	33	33
Transformed data shape	23138	22025	21503	21299	20762	20459
Transformed train set shape	16196	15417	15052	14909	14533	14321
Transformed test set shape	6942	6608	6451	6390	6229	6138

Table 4. Variation of accuracy of Machine learning algorithms in survival prediction in different Extreme Outlier handling windows of the proposed EOSRUS under-sampling technique

EOSRUS: Accuracy	**W=0.05**	**W=0.10**	**W=0.15**	**W=0.20**	**W=0.25**
Ada Boost Classifier	0.97	0.96	0.93	0.92	0.90
Decision Tree Classifier	0.96	0.96	0.93	0.92	0.90
Extra Trees Classifier	0.97	0.96	0.93	0.92	0.90
Extreme Gradient Boosting	0.97	0.96	0.93	0.92	0.90
Gradient Boosting Classifier	0.96	0.96	0.93	0.92	0.90
K Neighbors Classifier	0.95	0.95	0.92	0.91	0.89
Light Gradient Boosting Machine	0.97	0.96	0.93	0.93	0.90
Linear Discriminant Analysis	0.95	0.94	0.92	0.91	0.89
Logistic Regression	0.96	0.96	0.93	0.92	0.90
Naive Bayes	0.93	0.90	0.88	0.90	0.85
Quadratic Discriminant Analysis	0.92	0.81	0.80	0.90	0.76
Random Forest Classifier	0.96	0.96	0.93	0.92	0.90
Ridge Classifier	0.95	0.94	0.92	0.91	0.89
SVM - Linear Kernel	0.96	0.96	0.93	0.92	0.90

The experimental results from applying the Extreme Outlier Handled Stochastic Random Under-sampling (EOSRUS) algorithm show the classifier accuracy values across different extreme outlier handling window sizes (W) and various machine learning algorithms. These results deliver critical insights into how EOSRUS affects model performance in handling imbalanced datasets like VAERS.

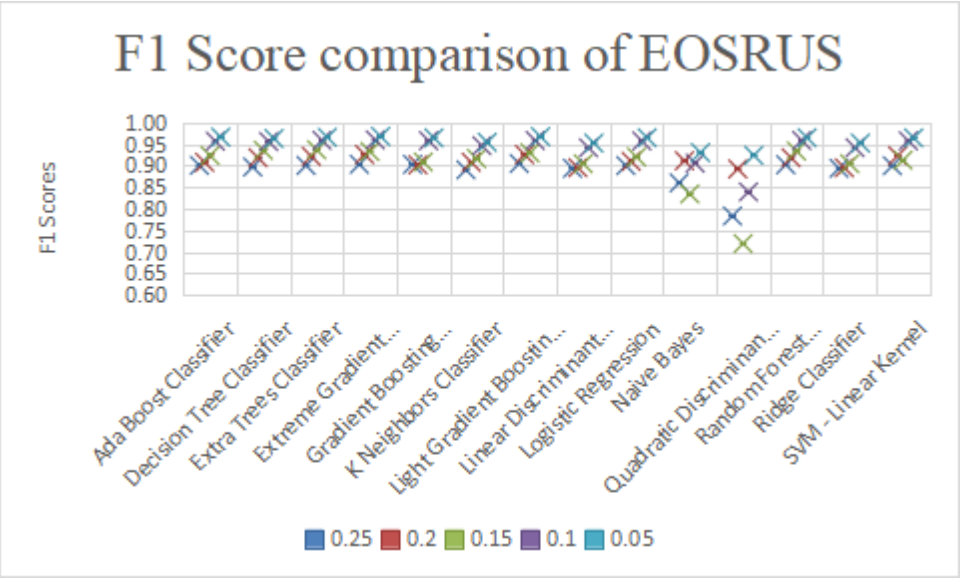
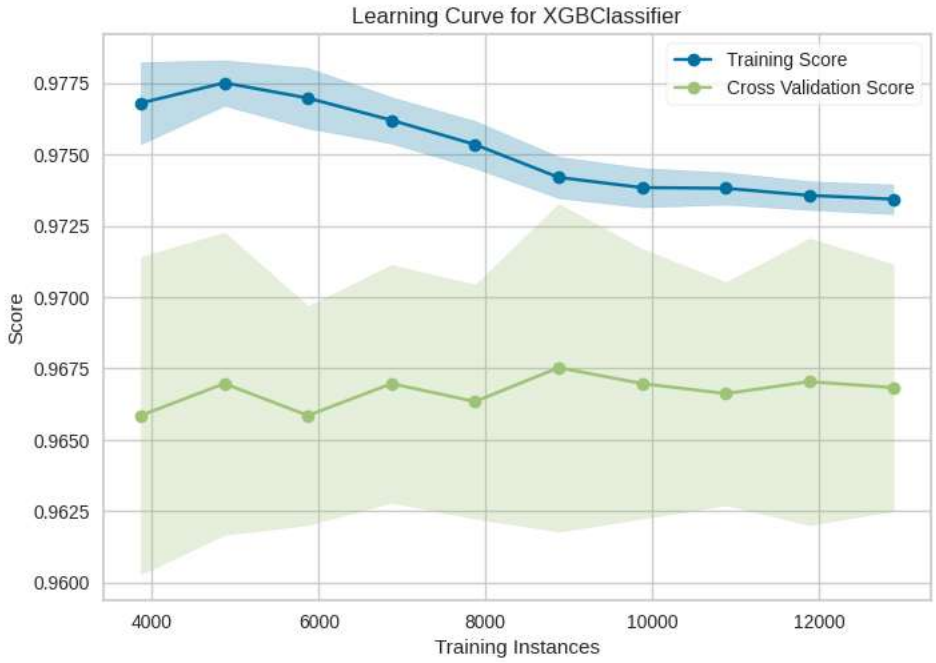


Fig.5 Comparison of F1 scores of various classification models in the different windows of EOSRUS

Among the various classifiers, XGBoost classifier has performed better with highest F1 score of 96.80%, hence the XG boost is fine tuned for better performance. In the Training process, there is a considerable reduction in delta between training and cross validation performance, indicating the increase in the class distribution contributing to better prediction performance. This is indicates that the algorithm neither underfit nor overfit during the prediction process.



Observations:
Effect of Outlier Handling Window (W) on Accuracy: The window W sets the threshold for defining “extreme outliers” in predicted class probabilities. Lower W values (e.g., 0.05) mean fewer samples are flagged as outliers and removed, leading to minimal data loss and higher accuracy overall. As W increases (up to 0.25), more samples are removed as outliers, which reduces training data size and can cause underfitting or loss of important information. This is reflected by a general decline in accuracy across classifiers as W grows due to loss of data while binning.
Classifier-Specific Performance Patterns: Ensemble classifiers such as AdaBoost, Extra Trees, Extreme Gradient Boosting, Gradient Boosting, and Random Forest sustained high accuracy (~0.96-0.97) at low W and showed a moderate decrease as W increased. Tree-based models and boosting methods demonstrated

robustness to under-sampling and outlier removal, possibly due to their ability to handle nonlinearities and variance through ensemble averaging. Parametric models such as Naive Bayes and Quadratic Discriminant Analysis (QDA) showed more significant accuracy drops at higher W, indicating greater sensitivity to data reduction and distributional assumptions. QDA fell as low as 0.76 accuracy at W=0.25. Models like Logistic Regression, SVM (Linear Kernel), and Ridge Classifier maintained relatively high accuracy with moderate sensitivity to W changes, offering stable baseline performance. The F1 Score comparison shows a similar pattern confirming the sensitivity to W changes across different classifiers.

Optimal Outlier Window Setting: The best accuracy and F1 Score for almost all classifiers occurs at W = 0.05, where only the most extreme misclassified samples are removed. This suggests a conservative outlier removal strategy best balances robustness and information preservation for VAERS data. Larger windows progressively remove more data points, likely pruning informative samples along with true outliers, damaging model performance.

Algorithm Ranking by Accuracy (at W=0.05) Highest accuracies (≥0.97): AdaBoost, Extra Trees, Extreme Gradient Boosting, LightGBM (all ensemble learners) obtained higher accuracy and F1 Score but slightly lower accuracy and F1 Score (~0.95-0.96) were noticed in Logistic Regression, SVM, Random Forest, Decision Tree, Ridge Classifier. Lowest accuracies (~0.92-0.93) was seen in QDA, Naive Bayes.

CONCLUSION

This study presents Extreme Outlier Handled Stochastic Random Undersampling (EOSRUS) as an effective approach to address the persistent challenge of class imbalance in VAERS dataset classification, specifically for predicting post-vaccination adverse events using bag-of-words features and traditional NLP. By integrating stochastic random under-sampling with targeted removal of extreme outliers samples highly likely to be misclassified based on classifier confidence EOSRUS improves the model's robustness and predictive performance. Experimental results across a diverse set of classifiers demonstrate that carefully tuning the extreme outlier handling window (W) enables a balanced trade-off between noise reduction and training data retention. Ensemble methods, particularly AdaBoost, Extra Trees, and Gradient Boosting variants, show superior accuracy and stability, benefiting most from EOSRUS at lower window thresholds (W = 0.05). In contrast, more sensitive parametric models experience performance degradation at higher W values, underscoring the importance of conservative outlier pruning. The evaluation confirms that EOSRUS enhances the overall predictive capability of machine learning models on highly imbalanced VAERS data, a critical factor in reliable post-vaccination adverse event detection. Importantly, this approach combines the computational efficiency of random under-sampling with principled outlier elimination, avoiding the pitfalls of synthetic oversampling techniques. Despite its demonstrated utility, the EOSRUS approach as applied to VAERS data with traditional NLP features has several important limitations:

- **Information Loss from Under-sampling:** Random under-sampling, even when combined with outlier removal, can discard informative majority-class examples. This may lead to underfitting or reduced generalizability, particularly when the discarded cases include borderline or complex samples critical for learning subtle decision boundaries.
- **Sensitivity to Window Hyperparameter (W):** Performance depends heavily on the correct choice of the outlier window (W). If set improperly, either too many or too few samples are removed, impairing the model's ability to learn from difficult instances or adequately suppressing noise. There is no universally optimal W, so extensive tuning is required for each context or dataset variant.
- **Classifier and Feature Dependency:** The method primarily boosts performance of ensemble models with bag-of-words features. Its effect may differ for deep learning architectures, transformer-based embeddings, or highly imbalanced multi-class targets, which were not explored in this work.
- **Computational Overhead from Iterative Procedure:** EOSRUS involves repeated (stochastic) under-sampling, model training, and outlier removal across multiple window settings and iterations, which can be computationally intensive for very large-scale or real-time datasets.
- **Potential for Over-Pruning:** Removing all extreme outliers assumes these are always true mislabels or noise, but in practice, some may be difficult-yet-informative cases. Excessive pruning risks biasing the model away from recognizing challenging adverse event patterns.
- **Applicability to Non-English or Non-Structured Data:** The method was only validated using structured, English-language VAERS data and may require adaptation for international pharmacovigilance databases or unstructured/noisy narrative fields.

FUTURE SCOPE

Integration with Deep NLP and Embedding Methods: Future work can examine how EOSRUS performs when the feature space is derived from contextualized embeddings (e.g., BERT, BioBERT) or hybrid representations. This might yield synergistic improvements for classifying complex textual reports.
Automated Window Tuning: Developing adaptive or meta-learning frameworks to optimize the outlier window (W) dynamically—possibly through Bayesian optimization or reinforcement learning—would generalize the method and reduce manual tuning overhead.

Application to Multi-Class & Hierarchical Outcomes: Extending and benchmarking EOSRUS for multi-class classification tasks (beyond binary outcomes) and rare-syndrome signal detection would enhance its utility in broader pharmacovigilance applications.

Explainability and Robustness Analysis: Investigating the interpretability of EOSRUS models, such as understanding which types of adverse events are most affected by outlier removal, and their robustness to adversarial or mislabeled samples, is an essential next step.

Adaptive Data Drift Handling: Adapting the method for streaming data environments, supporting continuous model updating, and explicitly addressing concept drift in VAERS-like datasets would increase its real-world applicability for ongoing vaccine safety monitoring.

Cross-Dataset and Cross-Region Validation: Evaluating EOSRUS across different countries' reporting systems or temporal slices of VAERS data would test its generalizability and inform best practices for global vaccine pharmacovigilance. In conclusion, while EOSRUS offers a practical and effective solution for imbalanced post-vaccination event classification in VAERS, these limitations and directions set the stage for future enhancements that will increase both the robustness and interpretability of automated vaccine safety analytics.

REFERENCES

1. Ahamad, Md. M., Aktar, S., Uddin, Md. J., Rashed-Al-Mahfuz, Md., Azad, A. K. M., Uddin, S., Alyami, S. A., Sarker, I. H., Khan, A., Liò, P., Quinn, J. M. W., & Moni, M. A. (2022). Adverse Effects of COVID-19 Vaccination: Machine Learning and Statistical Approach to Identify and Classify Incidences of Morbidity and Postvaccination Reactogenicity. Healthcare, 11(1), 31. https://doi.org/10.3390/healthcare11010031
2. Al Khames Aga, Q. A., Alkhaffaf, W. H., Hatem, T. H., Nassir, K. F., Batineh, Y., Dahham, A. T., Shaban, D., Al Khames Aga, L. A., Agha, M. Y. R., & Traqchi, M. (2021). Safety of COVID-19 vaccines. Journal of Medical Virology, 93(12), 6588–6594. https://doi.org/10.1002/jmv.27214
3. Albayati, M. B., & Altamimi, A. M. (2023). Analyzing COVID-19 Vaccine Adverse Reactions Using Machine Learning Techniques. Karbala International Journal of Modern Science, 9(2), 264–279. https://doi.org/10.33640/2405-609X.3271
4. Botsis, T., Nguyen, M. D., Woo, E. J., Markatou, M., & Ball, R. (2011). Text mining for the Vaccine Adverse Event Reporting System: medical text classification using informative feature selection. Journal of the American Medical Informatics Association, 18(5), 631–638. https://doi.org/10.1136/amiajnl-2010-000022
5. Du, J., Xiang, Y., Sankaranarayanapillai, M., Zhang, M., Wang, J., Si, Y., Pham, H. A., Xu, H., Chen, Y., & Tao, C. (2021). Extracting postmarketing adverse events from safety reports in the vaccine adverse event reporting system (VAERS) using deep learning. Journal of the American Medical Informatics Association, 28(7), 1393–1400. https://doi.org/10.1093/jamia/ocab014
6. Flora, J., Khan, W., Jin, J., Jin, D., Hussain, A., Dajani, K., & Khan, B. (2022). Usefulness of Vaccine Adverse Event Reporting System for Machine-Learning Based Vaccine Research: A Case Study for COVID-19 Vaccines. International Journal of Molecular Sciences, 23(15), 8235. https://doi.org/10.3390/ijms23158235
7. Gupta, H., & Verma, O. P. (2023). Vaccine hesitancy in the post-vaccination COVID-19 era: a machine learning and statistical analysis driven study. Evolutionary Intelligence, 16(3), 739–757. https://doi.org/10.1007/s12065-022-00704-3
8. Jeyanathan, M., Afkhami, S., Smaill, F., Miller, M. S., Lichty, B. D., & Xing, Z. (2020). Immunological considerations for COVID-19 vaccine strategies. Nature Reviews Immunology, 20(10), 615–632. https://doi.org/10.1038/s41577-020-00434-6
9. Kaur, S. P., & Gupta, V. (2020). COVID-19 Vaccine: A comprehensive status report. Virus Research, 288, 198114. https://doi.org/10.1016/j.virusres.2020.198114
10. Li, Y., Li, J., He, J., & Tao, C. (2024). AE-GPT: Using Large Language Models to extract adverse events from

surveillance reports-A use case with influenza vaccine adverse events. PLOS ONE, 19(3), e0300919. https://doi.org/10.1371/journal.pone.0300919
11. Okamoto, R., Kojima, R., & Nakatsui, M. (2023). Toward AI-supported Evaluation for Safety Control Measures against Near-Miss Events in Pharmaceutical Products. https://ssrn.com/abstract=4333919
12. Ravindranath, V., & Balakrishnan, S. (2023). COVID 19 post-vaccination adverse effects prediction with supervised machine learning models. 2023 International Conference on Computer Communication and Informatics (ICCCI), 1–6. https://doi.org/10.1109/ICCCI56745.2023.10128441
13. Riad, A., Schünemann, H., Attia, S., Peričić, T., Žuljević, M., Jürisson, M., Kalda, R., Lang, K., Morankar, S., Yesuf, E., Mekhemar, M., Danso-Appiah, A., Sofi-Mahmudi, A., Pérez-Gaxiola, G., Dziedzic, A., Apóstolo, J., Cardoso, D., Marc, J., Moreno-Casbas, M., … Klugar, M. (2021). COVID-19 Vaccines Safety Tracking (CoVaST): Protocol of a Multi-Center Prospective Cohort Study for Active Surveillance of COVID-19 Vaccines' Side Effects. International Journal of Environmental Research and Public Health, 18(15), 7859. https://doi.org/10.3390/ijerph18157859
14. Saad, E., Sadiq, S., Jamil, R., Rustam, F., Mehmood, A., Choi, G. S., & Ashraf, I. (2022). Novel extreme regression-voting classifier to predict death risk in vaccinated people using VAERS data. PLoS ONE, 17(6 June). https://doi.org/10.1371/journal.pone.0270327
15. Wunnava, S., Qin, X., Kakar, T., Kong, X., Rundensteiner, E. A., Sahoo, S. K., & De, S. (2018). One size does not fit all: An ensemble approach towards information extraction from adverse drug event narratives. HEALTHINF 2018 - 11th International Conference on Health Informatics, Proceedings; Part of 11th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2018, 5, 176–188. https://doi.org/10.5220/0006600201760188
16. Li, Y., Viswaroopan, D., He, W., Li, J., Zuo, X., Xu, H., & Tao, C. (2025). Improving entity recognition using ensembles of deep learning and fine-tuned large language models: A case study on adverse event extraction from VAERS and social media. Journal of Biomedical Informatics, 163, 104789. https://doi.org/10.1016/j.jbi.2025.104789
17. Chen, N., Sun, Z., & Jia, T. (2022). A comparative study on application of class-imbalance learning for severity prediction of adverse events following immunization. arXiv preprint arXiv:2206.09752. https://arxiv.org/abs/2206.09752
18. Tiwari, A., Bolla, B. K., & Bonthu, S. (2024). Machine learning and NLP approach to predict hospitalization upon adverse drug reaction symptoms of COVID-19 vaccine administration. In Artificial Intelligence and Knowledge Processing: AIKP 2023, Communications in Computer and Information Science (Vol. 2127, pp. 372–384). Springer, Cham. https://doi.org/10.1007/978-3-031-68617-7_25
19. Abuzeid, A., Jolkver, E., & Li, H. (2026). Rare event detection by progressive clustering undersampling. PLOS ONE, 21(1), e0340758. https://doi.org/10.1371/journal.pone.0340758